

# COURSE OUTLINE

## ASP .NET Core

Learners Point  
ACADEMY

The Business Center - Suite 610 -  
Khalid Bin Al Waleed Rd - Al Hamriya  
- Dubai - United Arab Emirates

## Course Overview

This comprehensive ASP.NET Core course provides a deep dive into building, securing, and deploying robust APIs using the ASP.NET Core framework. Participants will start by getting acquainted with the ASP.NET Core environment, including project setup, the request pipeline, and middleware configuration. The course covers essential skills such as creating and managing APIs, handling routing, and returning various types of resources with correct status codes and error responses. Learners will gain expertise in input validation, dependency injection, and logging, as well as in using Entity Framework Core for data management. Advanced topics include implementing security with token-based authentication and OAuth2, versioning and documenting APIs with Swagger, and performing comprehensive testing. The course culminates in deploying APIs to Azure App Service and configuring advanced features like Application Insights and Azure Key Vault. Through practical labs and demonstrations, participants will develop the skills needed to build secure, scalable, and well-documented APIs ready for production environments.

## Course Schedule

Duration : 3 Days

## Module 1: Getting Acquainted with ASP.NET Core

- Frameworks and Tooling
- ASP.NET Core: The Big Picture
- Approaches to Building APIs with ASP.NET Core
- Creating and Running a New ASP.NET Core Project
- Running an ASP.NET Core Project Using the CLI
- Inspecting the ASP.NET Core API Project Code Structure
- The ASP.NET Core Request Pipeline & Middleware
- Add Swagger Middleware
- Configuring the ASP.NET Core Request Pipeline
- Working with Different Environments
- Add Swagger Development Only

## Module 2: Creating the API and Returning Resources

- Registering API Services on the Container
- Returning Resources
- Learning About Routing
- Add HttpGet Attribute
- Using Postman
- Add Route Attribute

- Improving the Architecture with Model Classes
- The Importance of Status Codes
- Returning Correct Status Codes
- Returning Child Resources
- Describing Error Responses with Problem Details
- Manipulating Problem Details Responses
- Return Not Found
- Formatters and Content Negotiation
- Getting a File
- Add XML Formatters
- Return File

## Module 3: Manipulating Resources and Validating Input

- Passing Data to the API
- Creating a Resource
- Use Binding Source Attribute
- Validating Input
- A Validation Alternative
- Add Validation Attribute
- Updating a Resource
- Partially Updating a Resource
- Adding Support for JsonPatchDocument
- Deleting a Resource
- Creating a File 5m

## Module 4: Working with Services and Dependency Injection

- Inversion of Control and Dependency Injection
- Injecting and Using a Logger
- Logging Exceptions
- Change Log Level 0m Demo:
- Globally Handling and Logging Exceptions
- Replacing the Default Logger and Logging to a File 6m
- Implementing and Using a Custom Service
- Registering a Service by Interface
- Working with Configuration Files
- Scoping Configuration to Environments

## Module 5: Getting Acquainted with Entity Framework Core

- Introducing Entity Framework Core
- Creating Entity Classes Demo
- Creating a DbContext
- Working with Migrations
- Seeding the Database with Data
- Safely Storing Sensitive Configuration Data
- SQL Injection
- Implementing Joins, grouping etc.



## Module 6: Using Entity Framework Core in Your Controllers • Introducing the Repository Pattern

- The Purpose of Asynchronous Code
- Add Repository Method
- Returning Data from the Repository When Requesting Resources
- Using AutoMapper to Map Between Entities and DTOs
- Register AutoMapper Demo: Creating a Resource
- Create AutoMapper Map
- Updating a Resource
- Partially Updating a Resource
- Deleting a Resource

## Module 7: Searching, Filtering, and Paging Resources

- Searching and Filtering Resources
- Accept Query String Value
- Searching Through Resources
- Deferred Execution
- Implement Search Query
- Paging Through Resources
- Returning Pagination Metadata
- Returning Pagination Metadata 4m

## Module 8: Securing Your API

- A Few Words on Securing APIs
- Implementing Token-based Security
- Creating a Token
- Requiring and Validating a Token
- Using Information from the Token in Your Controller
- Working with Authorization Policies
- Using Information from the Token in an Authorization Policy
- Generating a Token with dotnet user-jwts
- Improving Token-based Security with OAuth2 and OpenID Connect

## Module 9: Versioning and Documenting Your API

- Versioning Your API
- Supporting Versioning
- Add Versioned Route
- Supporting Versioned Routes
- Documenting Your API with OpenAPI / Swagger
- Adding Swagger Support to a Project
- Incorporating XML Comments on Actions
- Describing Response Types and Status Codes
- Supporting Different Documentation Versions
- Documenting API Authentication

## Module 10: Testing and Deploying Your API

- Testing API Endpoints
- Installing the HTTP\_REPL
- Supporting Authenticated Requests When Testing with the HTTP\_REPL
- Testing with .http Files
- Supporting Authenticated Requests When Testing with .http Files
- Facilitating API testing with Endpoints Explorer
- Hosting and Deploying an ASP.NET Core API
- Considering Your Deployment and Hosting Options
- Dealing with Proxies and Load Balancers
- Deploying to an Azure App Service
- Configuring Logging on Application Insights
- Improving Storing Secrets with Azure Key Vault

## Program Outcomes

Upon completing this ASP.NET Core course, participants will be adept at designing, building, and deploying secure and scalable APIs using the ASP.NET Core framework. They will possess a thorough understanding of API development fundamentals, including setting up projects, configuring middleware, and implementing routing and resource management. Participants will be skilled in handling input validation, utilizing dependency injection, and integrating Entity Framework Core for data access. They will also be proficient in applying security measures such as token-based authentication and OAuth2, as well as versioning and documenting APIs with Swagger. Additionally, they will have practical experience in testing API endpoints, deploying applications to Azure App Service, and managing configuration and secrets using Azure Key Vault. Overall, participants will be prepared to develop high-quality APIs that are secure, well-documented, and ready for production deployment.